

Proof Nets for Display Logic

Richard Moot

July 6, 2007

1 Introduction

Moot & Puite (2002) have introduced proof nets for the multimodal Lambek calculus $\mathbf{NL}\diamond_{\mathcal{R}}$. Since then, numerous other connectives have been proposed to deal with different linguistic phenomena, a par — or co-tensor, as some authors prefer to call it — together with the corresponding co-implication (Lambek 1993, Moortgat 2007, Bernardi & Moortgat 2007), Galois and dual-Galois connectives (Areces, Bernardi & Moortgat 2001).

We can incorporate these extensions (as well as a few others) into the proof net calculus by simply dropping the restriction that sequents are trees with a unique root node and obtain what are, in effect, proof nets for display logic (Goré 1998). The notion of contraction generalizes to these new connectives without complications.

Like for the Lambek calculus, proof nets for display logic have the advantage of collapsing proofs which differ only for trivial reasons. The display rules in particular are compiled away in the proof net representation.

2 Proof Nets

Proof nets are an optimal representation for proofs of linear logic introduced by Girard (1987).

2.1 Links and Proof Structures

Definition 1 A link, as defined by Moot & Puite (2002)¹, is a tuple $\langle \tau, P, Q, m \rangle$ where

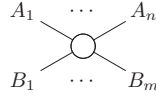
- τ , the type of the link, is either \otimes or \wp ,
- P is a list of premisses A_1, \dots, A_n ,

¹Some of the details are slightly different: the rule name ν has been suppressed since we need only the mode part of it and the subsequences p and q have been replaced by the main formula argument m

- Q is a list of conclusions B_1, \dots, B_m ,
- m , the main formula of the link, is either ϵ or a member of $P \cup Q$.

If $m = \epsilon$ then we will call the link *neutral*, if it is a member of P we will call the link a *left link* and if it is a member of Q we will call it a *right link*.

We draw links as shown below, with the premisses from left to right above the link and the conclusions below it.



Visually, we distinguish between tensor links — which we draw with a white circle at the interior — and par links — which are drawn with a black circle. Finally, unless $m = \epsilon$ we denote the main formula of the link by drawing an arrow from the center of the link to this formula. In this case, we will refer to the other formulas as the *active formulas* of the link.

This definition of link allows us to create quite a number of links in addition to the ones given in that article. The links there were all possible unary and binary links given the assumption of a unique conclusion for every tensor link. Once we drop this constraint, different types of link become possible.

Figure 1 gives an overview of the 9 different forms of tensor links of arity 2 or less (2 nullary, 3 unary and 4 binary), together with the logical connectives associated with their different ports for a total of 2 nullary, 6 unary and 12 binary connectives.

Note that — as displayed in the figure — none of the tensor links have a main formula according to Definition 1. However, in case we need to find the main and active formulas of a link, we can do so by simply inspecting the formulas assigned to the different ports.

Corresponding to each tensor link is one par link which is a ‘mirror image’ of the corresponding tensor link as shown in Figure 2

If we want to make more distinctions, we can use modes — as is usual in multimodel categorial grammar (Moortgat 1997) — as we did in (Moot & Puite 2002) for $\mathbf{NL}_{\diamond \mathcal{R}}$ and write the mode in the circle of the link. To somewhat reduce the (already extensive) vocabulary, we will not talk about modes in this article, but the current approach can be extended to incorporate them without problems. Adding them would just amount to inserting mode information in all tensor and par links and demanding identity between the two modes to allow a contraction.

Definition 2 A proof structure $\langle S, \mathcal{L} \rangle$ is a finite set of formulas S together with a set of links \mathcal{L} as shown in Figures 1 and 2 such that.

- every formula of S is at most once the premiss of a link.

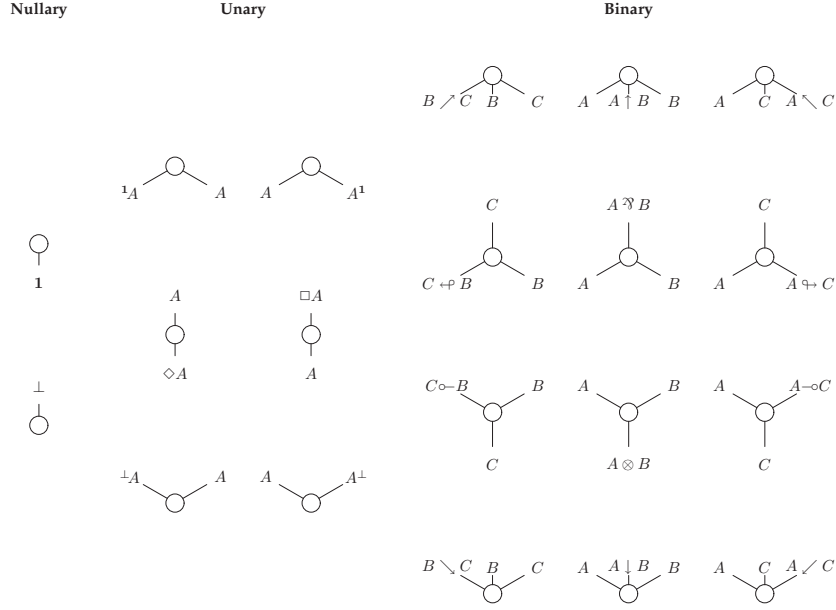


Figure 1: All tensor links of arity 2 or less

- every formula of S is at most once the conclusion of a link.

Formulas which are not the conclusion of any link are the hypotheses H of the proof structures, whereas the formulas which are not the premiss of any link are the conclusions C .

Readers familiar with proof nets from linear logic will note the absence or cut and axiom links. We have axiom and cut *formulas* instead.

Definition 3 An axiom formula is a formula which is not the main formula of any link. A cut formula is a formula which is the main formula of two links.

Figure 3 shows the proof structure for $(\perp \multimap A) \multimap \perp \vdash A$ on the left. The A formula is the only axiom in the structure.

There are some differences in the notation of other authors. It is closest to display logic, with \wp taking the place of \oplus and \perp taking the place of 0 , much in the spirit of the connectives from linear logic. The symbols for the two implications \multimap and \multimap have been chosen to remind us they are the residuals of \wp . Table 7 gives an overview of the logical symbols used and the corresponding logical symbols in various other logics.

2.2 Abstract Proof Structures

From a proof structure we obtain an *abstract* proof structure simply by erasing all formulas on the internal nodes. We only keep the formulas on the premisses

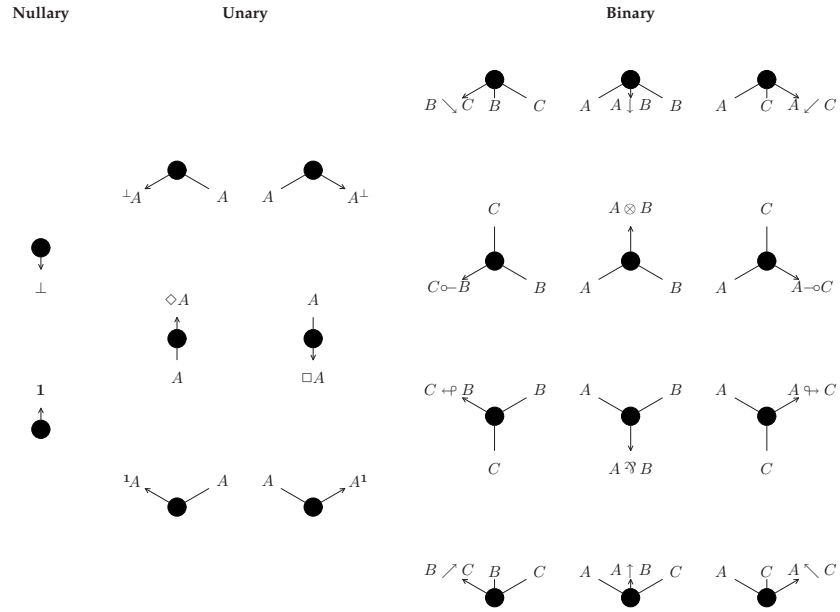


Figure 2: All par links of arity 2 or less

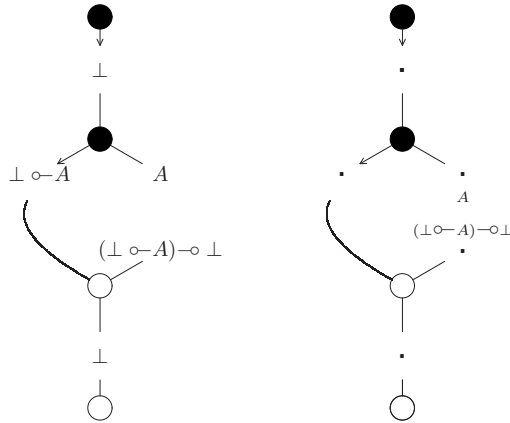


Figure 3: Proof structure and abstract proof structure

and conclusions of the proof structure, ie. just the leaves.

Definition 4 An abstract proof structure is a tuple $\langle V, \mathcal{L}, p, q \rangle$ such that.

V is a finite set of vertices,

\mathcal{L} is a set of links such that

- every vertex of V is at most once the premiss of a link,
- every vertex of V is at most once the conclusion of a link

p is a labelling function assigning a formula to the hypotheses of the abstract proof structure, that is, to those formulas which are not the conclusion of any link,

q is a labelling function assigning a formula to the conclusions of the abstract proof structure, that is, to those formulas which are not the premiss of any link.

We will draw the nodes of abstract proof structures as shown below

$$\begin{array}{c} H \\ \bullet \\ C \end{array}$$

where H is the hypothesis assigned to this node and C is the conclusion assigned to it. Both H and C can be empty.

Figure 3 shows the abstract proof structure corresponding to the proof structure of $(\perp \multimap A) \multimap \perp \vdash A$ on the right.

Definition 5 A tensor tree is an acyclic connected abstract proof structure containing only tensor links.

We say a tensor tree with hypotheses A_1, \dots, A_n and conclusions B_1, \dots, B_m corresponds to $A_1, \dots, A_n \vdash B_1, \dots, B_m$. However, in order to determine the structure of the sequent to which a tensor tree corresponds, we first have to do a bit of work.

2.3 Sequents and Tensor Trees

An advantage of the formulation of Moot & Puite (2002) was that, because of the shape of the two tensor links we considered and because of the conditions on proof structures, a tensor tree was a rooted tree. The new types of tensor links do not preserve this property. Figure 4 shows an example.

Here, we have three premisses (A , B and C) and two conclusions (D and E) but they are grouped in such a way that we cannot turn them into a sequent $A, B, C \vdash D, E$ straightforwardly.

To solve this problem, we abolish the notion that the premisses of a sequent are on the left hand side of the turnstile and the conclusions on the right hand side. We simply split the tensor tree at an arbitrary point and translate the two trees we obtain into sequents in such a way that we can recover the original tensor tree.

Figure 5 lists the structural connectives we need: 1 nullary, 3 unary and 6 binary. The structural connectives are essentially borrowed from display logic.

Definition 6 Let T be a tensor tree and x be a node on this tensor tree, the sequent $T(x)$ is defined as follows. We split T at x to obtain a tree T_h^x which has x as a hypothesis and a tree T_c^x which has x as a conclusion. Without changing the shape of

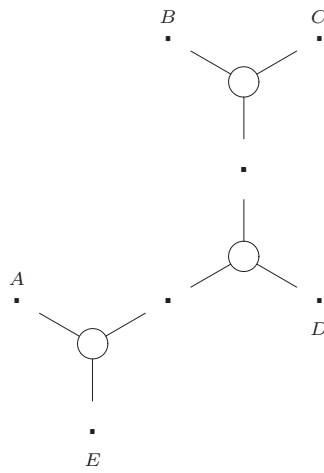


Figure 4: A tensor tree which is not rooted

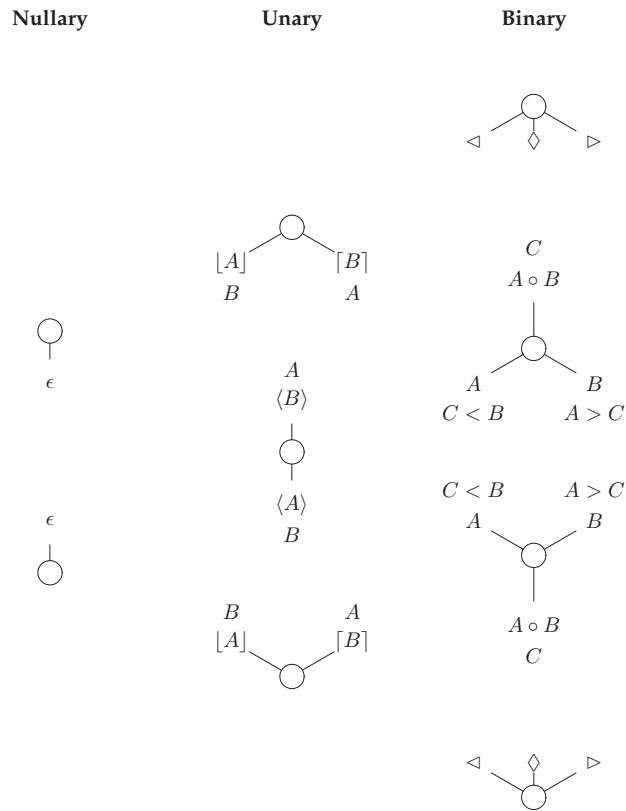


Figure 5: Information Flow

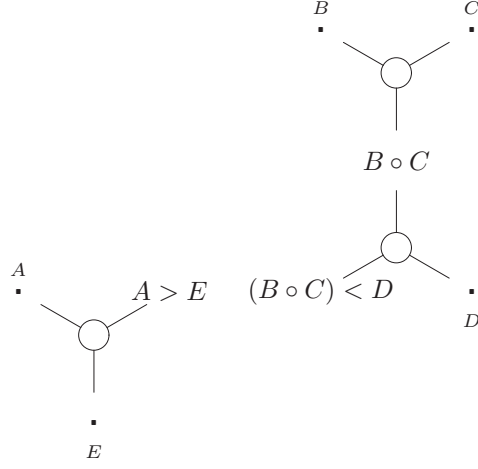


Figure 6: Computing the flow

$$\begin{array}{c}
 \frac{[\Gamma] \vdash \Delta}{[\Delta] \vdash \Gamma} [dgc] \quad \frac{\Gamma \vdash [\Delta]}{\Delta \vdash [\Gamma]} [gc] \\
 \\
 \frac{\langle \Gamma \rangle \vdash \Delta}{\Gamma \vdash \langle \Delta \rangle} [rc] \\
 \\
 \frac{\Gamma_1 \vdash \Delta < \Gamma_2}{\Gamma_1 \circ \Gamma_2 \vdash \Delta} [rc] \quad \frac{\Gamma < \Delta_2 \vdash \Delta_1}{\Gamma \vdash \Delta_1 \circ \Delta_2} [drc] \\
 \frac{\Gamma_2 \vdash \Gamma_1 > \Delta}{\Gamma_2 \vdash \Gamma_1 > \Delta} [rc] \quad \frac{\Delta_1 > \Gamma \vdash \Delta_2}{\Delta_1 > \Gamma \vdash \Delta_2} [drc] \\
 \\
 \frac{\Delta \triangleright \Gamma_2 \vdash \Gamma_1}{\Gamma_1 \diamond \Gamma_2 \vdash \Delta} [dgc] \quad \frac{\Delta_1 \vdash \Gamma \triangleleft \Delta_2}{\Gamma \vdash \Delta_1 \diamond \Delta_2} [gc] \\
 \frac{\Gamma_1 \triangleright \Delta \vdash \Gamma_2}{\Gamma_1 \triangleright \Delta \vdash \Gamma_2} [dgc] \quad \frac{\Delta_2 \vdash \Delta_1 \triangleright \Gamma}{\Delta_2 \vdash \Delta_1 \triangleright \Gamma} [gc]
 \end{array}$$

Table 1: Sequent Rules — Display Rules

either of the trees, we will consider the two instances x as the root of their respective trees and all other hypotheses and conclusions as its leaves. Moving from these leaves towards x we use the flow of Figure 5 to compute a term S_c for T_c^x and a term S_h for T_h^x . The final sequent $T(x)$ is $S_c \vdash S_h$.

Note that the tree upwards of the split point becomes the antecedent, while the tree down from it becomes the succedent. Figure 6 shows an example of computing the flow corresponding a split vertex.

We see that, depending on our choice of the ‘split point’ of the tensor tree, Figure 4 corresponds to one of the following sequents, of which we computed the second in Figure 6.

$$\begin{aligned}
& A \vdash E < ((B \circ C) < D) \\
& (B \circ C) < D \vdash A > E \\
& A \circ ((B \circ C) < D) \vdash E \\
& B \circ C \vdash (A > E) \circ D \\
& B \vdash ((A > E) \circ D) < C \\
& C \vdash B > ((A > E) \circ D) \\
& (A > E) > (B \circ C) \vdash D
\end{aligned}$$

There is exactly one possible sequent for each vertex in the graph; this is no coincidence as it corresponds to a ‘display property’ for each vertex. Note that all of these sequents are interderivable thanks to the display rules of Table 1. In the following, to make it easier to refer to each of the display rules, we will write the two structural connectives between parentheses. For example, we will write $rc(<; >)$ for a replacement (from top to bottom) for a $<$ structural connective by a $>$ structural connective.

Lemma 7 *Let \mathcal{T} be a tensor tree and x and y two nodes on this tree. Take $s = \mathcal{T}(x)$ and $s' = \mathcal{T}(y)$. s and s' are equivalent up to the display rules.*

Proof Induction on the length l of the unique path between x and y . In case $l = 0$ then s and s' are identical.

Suppose $l > 0$, induction hypothesis ... we essentially replace one structural connective by another corresponding to either $[rc]$, $[drc]$, $[gc]$ or $[rc]$... \square

Lemma 8 *If \mathcal{T} is a tensor tree containing one or more links, then at least one of the leaves of the corresponding proof structure (hypotheses and conclusions) is the main formula of its link.*

Proof Follow the arrows: start at an arbitrary link, if the main formula of this link is not a leaf, then it must be the active formula of another link (if it would be the main formula of another link, this other link is a par link, contradicting that \mathcal{T} is a tensor tree), etc. \square

2.4 Sequent Rules

In addition to the display rules, which allow us to turn any formula in the sequent to either the complete left hand side or the complete right hand side of a sequent, we have a left and right rule for each of the connectives.

Given this property we can always assume — as shown by the rules in Tables 2 to 5 — that the context is on the other side of the sequent as the logical connective we would want to treat. Apart from the binary galois and dual galois connectives, which I haven’t seen elsewhere, these rules are the same up to notational choices as those of display logic.

$$\frac{\epsilon \vdash \Delta}{\mathbf{1} \vdash \Delta} [L\mathbf{1}] \quad \frac{}{\epsilon \vdash \mathbf{1}} [R\mathbf{1}]$$

$$\frac{}{\perp \vdash \epsilon} [L\perp] \quad \frac{\Gamma \vdash \epsilon}{\Gamma \vdash \perp} [R\perp]$$

Table 2: Sequent Rules — Nullary Connectives

$$\frac{\Delta \vdash A}{A^\perp \vdash [\Delta]} [L.\perp] \quad \frac{\Gamma \vdash [A]}{\Gamma \vdash A^\perp} [R.\perp]$$

$$\frac{\Delta \vdash A}{\perp A \vdash [\Delta]} [L^\perp.] \quad \frac{\Gamma \vdash [A]}{\Gamma \vdash \perp A} [R^\perp.]$$

$$\frac{[A] \vdash \Delta}{A^\mathbf{1} \vdash \Delta} [L.\mathbf{1}] \quad \frac{A \vdash \Gamma}{[\Gamma] \vdash A^\mathbf{1}} [R.\mathbf{1}]$$

$$\frac{[A] \vdash \Delta}{\mathbf{1} A \vdash \Delta} [L^\mathbf{1}.] \quad \frac{A \vdash \Gamma}{[\Gamma] \vdash \mathbf{1} A} [R^\mathbf{1}.]$$

$$\frac{\langle A \rangle \vdash \Delta}{\diamond A \vdash \Delta} [L\diamond] \quad \frac{\Gamma \vdash A}{\langle \Gamma \rangle \vdash \diamond A} [R\diamond]$$

$$\frac{A \vdash \Delta}{\square A \vdash \langle \Delta \rangle} [L\square] \quad \frac{\Gamma \vdash \langle A \rangle}{\Gamma \vdash \square A} [R\square]$$

Table 3: Sequent Rules — Unary Connectives

2.5 Contractions

A tensor and a par link contract when the tensor link is connected — respecting up/down and left/right — to the par link at all its ports except the single main port of the par link and the corresponding port of the tensor link. Note that the nullary connectives satisfy this demand trivially.

The redex for all contraction is a single node as follows.

$$\begin{array}{c} H \\ \bullet \\ C \end{array}$$

Both links and the internal nodes will be removed from the resulting graph and the two exterior nodes will be merged, inheriting the hypothesis and conclusion label of the nodes in case either node is a hypothesis or conclusion of the abstract proof structure.

Figure 7 shows the contractions for the binary residuated connectives \multimap , \otimes and \multimap . These are exactly the contractions proposed for NL.

Figure 8 shows their duals: the contractions for \multimap , \otimes and \multimap .

Figures 9 and 10 show the unary contractions for the Galois, residuated and dual Galois connectives.

$\frac{A \circ B \vdash \Delta}{A \otimes B \vdash \Delta} [L\otimes]$	$\frac{\Gamma_1 \vdash A \quad \Gamma_2 \vdash B}{\Gamma_1 \circ \Gamma_2 \vdash A \otimes B} [R\otimes]$
$\frac{\Gamma \vdash A \quad B \vdash \Delta}{A \multimap B \vdash \Gamma > \Delta} [L\multimap]$	$\frac{\Gamma \vdash A > B}{\Gamma \vdash A \multimap B} [R\multimap]$
$\frac{A \vdash \Delta \quad \Gamma \vdash B}{A \circ\!-\! B \vdash \Delta < \Gamma} [L\circ\!-\!]$	$\frac{\Gamma \vdash A < B}{\Gamma \vdash A \circ\!-\! B} [R\circ\!-\!]$
$\frac{A \vdash \Delta_1 \quad B \vdash \Delta_2}{A \wp B \vdash \Delta_1 \circ \Delta_2} [L\wp]$	$\frac{\Gamma \vdash A \circ B}{\Gamma \vdash A \wp B} [R\wp]$
$\frac{A > B \vdash \Delta}{A \wp\!-\! B \vdash \Delta} [L\wp\!-\!]$	$\frac{A \vdash \Delta \quad \Gamma \vdash B}{\Delta > \Gamma \vdash A \wp\!-\! B} [R\wp\!-\!]$
$\frac{A < B \vdash \Delta}{A \wp\!-\! B \vdash \Delta} [L\wp\!-\!]$	$\frac{\Gamma \vdash A \quad B \vdash \Delta}{\Gamma < \Delta \vdash A \wp\!-\! B} [R\wp\!-\!]$

Table 4: Sequent Rules — Binary Connectives

$\frac{\Gamma_1 \vdash A \quad \Gamma_2 \vdash B}{A \downarrow B \vdash \Gamma_1 \diamond \Gamma_2} [L\downarrow]$	$\frac{\Gamma \vdash A \diamond B}{\Gamma \vdash A \downarrow B} [R\downarrow]$
$\frac{\Gamma_1 \vdash A \quad \Gamma_2 \vdash B}{A \swarrow B \vdash \Gamma_1 \triangleleft \Gamma_2} [L\swarrow]$	$\frac{\Gamma \vdash A \triangleleft B}{\Gamma \vdash A \swarrow B} [R\swarrow]$
$\frac{\Gamma_1 \vdash A \quad \Gamma_2 \vdash B}{A \searrow B \vdash \Gamma_1 \triangleright \Gamma_2} [L\searrow]$	$\frac{\Gamma \vdash A \triangleright B}{\Gamma \vdash A \searrow B} [R\searrow]$
$\frac{A \diamond B \vdash \Delta}{A \uparrow B \vdash \Delta} [L\uparrow]$	$\frac{A \vdash \Delta_1 \quad B \vdash \Delta_2}{\Delta_1 \diamond \Delta_2 \vdash A \uparrow B} [R\uparrow]$
$\frac{A \triangleleft B \vdash \Delta}{A \nwarrow B \vdash \Delta} [L\nwarrow]$	$\frac{A \vdash \Delta_1 \quad B \vdash \Delta_2}{\Delta_1 \triangleleft \Delta_2 \vdash A \nwarrow B} [R\nwarrow]$
$\frac{A \triangleright B \vdash \Delta}{A \nearrow B \vdash \Delta} [L\nearrow]$	$\frac{A \vdash \Delta_1 \quad B \vdash \Delta_2}{\Delta_1 \triangleright \Delta_2 \vdash A \nearrow B} [R\nearrow]$

Table 5: Sequent Rules — Binary Connectives (continued)

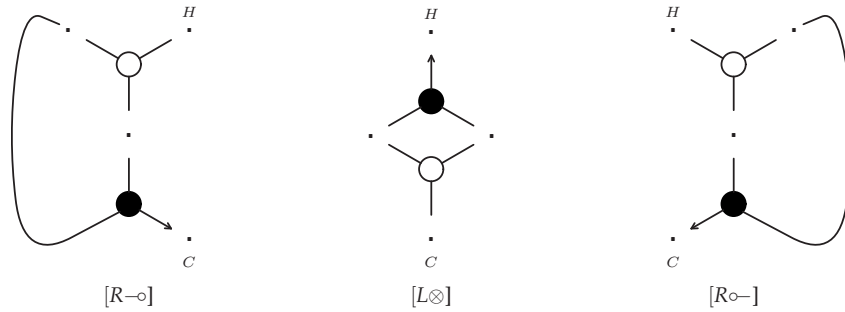


Figure 7: Contractions: binary residuated

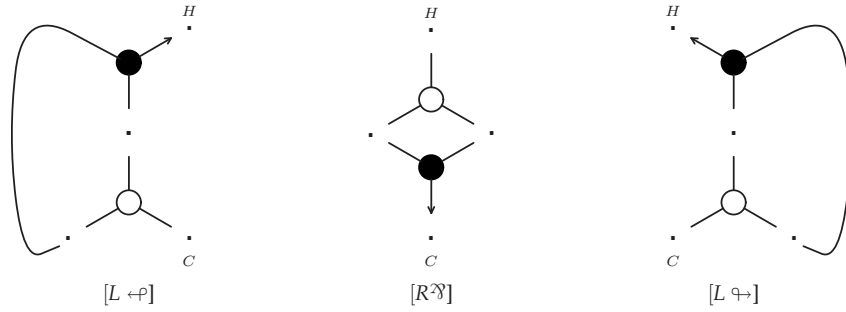


Figure 8: Contractions: binary dual residuated

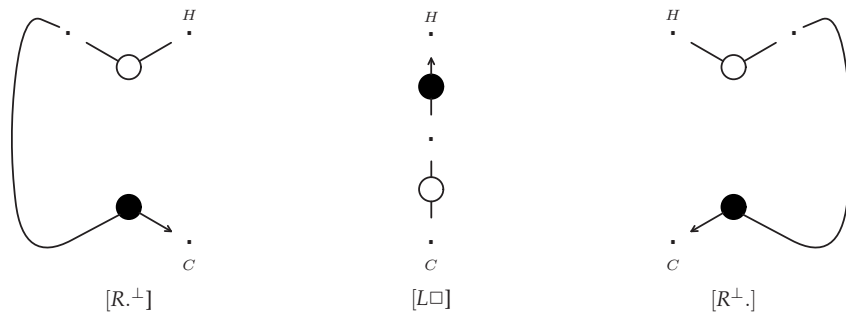


Figure 9: Contractions: unary 1

2.6 Structural Rules

We can extend the proof net as well as the sequent calculus with an arbitrary number of structural conversions. A structural conversion in the proof net cal-

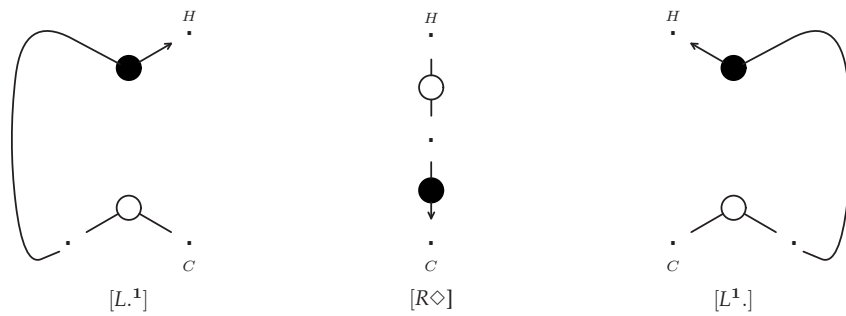


Figure 10: Contractions: unary 2

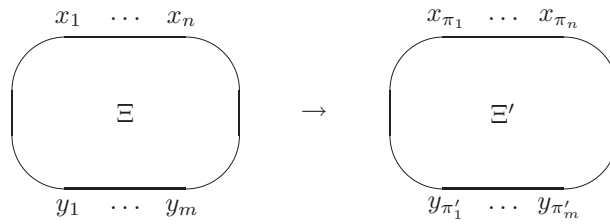


Figure 11: Schematic Form of a Structural Conversion

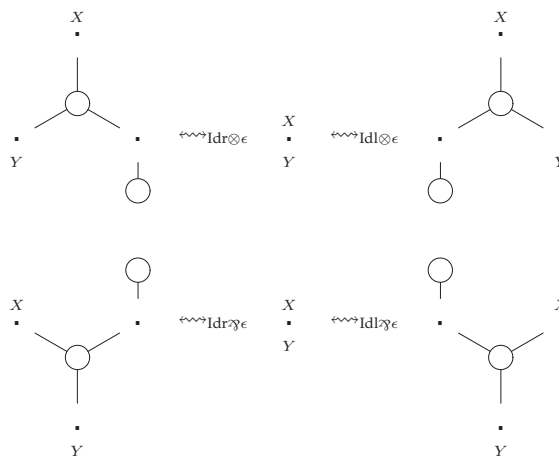


Figure 12: Identity rules for tensor and par

culus is simply a rewrite of one tensor tree into another in such a way that the both trees have the same hypotheses and the same conclusions, though we are allowed to change their order. Figure 11 shows the schematic form of a structural conversion. The x vertices are the n hypotheses of the conversion, the y vertices the m conclusions and π (resp. π') is a permutation of the hypotheses (resp. the conclusions).

This restriction means the contraction and weakening rules are not allowed: we operate essentially in a fragment of multiplicative linear logic (Girard 1987).

Figures 12 and 13 shows some well-known examples of valid structural rules which don't change the order of the hypotheses and premisses.

The structural rules of Figure 13 play a role similar to the mixed associativity rules in multimodal system. The corresponding mixed commutativity rules are shown in Figure 14.

Given a structural conversion, what is the sequent rule which corresponds to it? As shown by the following lemma, there are multiple equivalent possibilities, depending on which of the leaves is displayed.

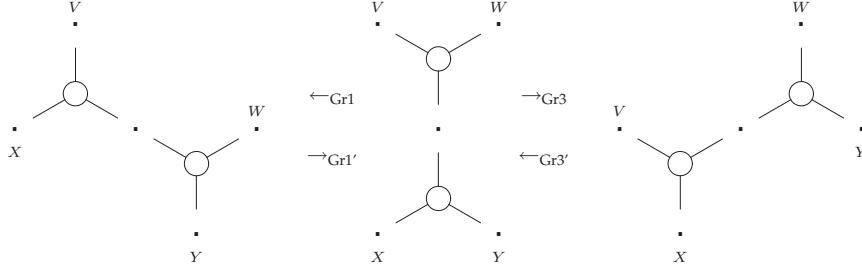


Figure 13: Grishin Rules: Mixed Associativity

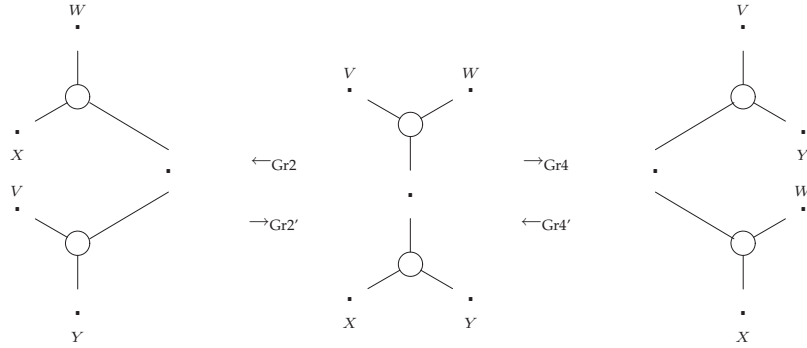


Figure 14: Grishin Rules: Mixed Commutativity

Definition 9 Let s be a structural conversion and l one of its leaves. $s(l)$ will denote the structural rule obtained by computing the flow according to Definition 6 with the exception that every hypothesis leaf x_i will correspond to a structural variable Γ_i and every conclusion leaf y_i to a structural variable Δ_i .

For example, depending on whether we use hypothesis X or conclusion Y to obtain a corresponding structural rule, we obtain either rule $[Idr \otimes \epsilon 1]$ or rule $[Idr \otimes \epsilon 2]$.

$$\frac{\Gamma \circ \epsilon \vdash \Delta}{\Gamma \vdash \Delta} [Idr \otimes \epsilon 1] \qquad \frac{\Gamma \vdash \Delta < \epsilon}{\Gamma \vdash \Delta} [Idr \otimes \epsilon 2]$$

Note that the two rules are equivalent.

$$\frac{\Gamma \vdash \Delta < \epsilon}{\Gamma \circ \epsilon \vdash \Delta} [rc < \circ] \qquad \frac{\Gamma \circ \epsilon \vdash \Delta}{\Gamma \vdash \Delta < \epsilon} [rc < \circ]$$

$$\frac{\Gamma \circ \epsilon \vdash \Delta}{\Gamma \vdash \Delta} [Idr \otimes \epsilon 1] \qquad \frac{\Gamma \vdash \Delta < \epsilon}{\Gamma \vdash \Delta} [Idr \otimes \epsilon 2]$$

Lemma 10 Let l_1 and l_2 be two distinct leaves of a structural conversion s . Then the structural rules $s(l_1)$ and $s(l_2)$ are interderivable using only the other rule and the display rules.

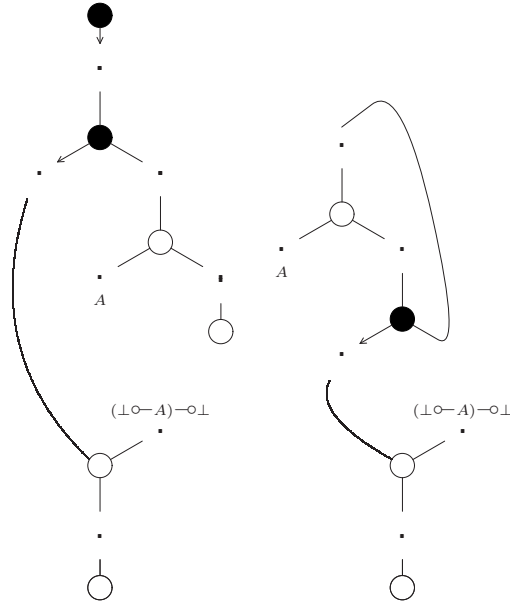


Figure 15: Right identity for \otimes , followed by the \perp contraction

Proof Similar to the proof of Lemma 7 we follow the unique path from l_1 to l_2 applying a display rule at each step to derive $s(l_2)$ from $s(l_1)$. Given that all the display rules are reversible we can derive $s(l_1)$ from $s(l_2)$ using the inverse rules. \square

Example 11 Using $Gr1'$, $Gr1$ and the right identity for tensor and left identity for par, we can derive $(\perp \circ - A) \circ \perp \vdash A$. Figures 15 to 18 show how the abstract proof structure of Figure 3 can be contracted.

3 Correctness

We are now in a position to prove the main theorem: that derivability in the sequent calculus and contractability in the proof net calculus coincide.

Theorem 12 A proof structure \mathcal{S} is correct (ie. corresponds to a sequent proof of $\Gamma \vdash \Delta$) if and only if its abstract proof structure \mathcal{A} converts to a tensor tree of $\Gamma \vdash \Delta$.

\Rightarrow Suppose π is a sequent calculus proof of $\Gamma \vdash \Delta$. We construct a proof structure together with a reduction sequence ρ reducing it to tensor tree $\Gamma \vdash \Delta$ by induction on the depth d of π .

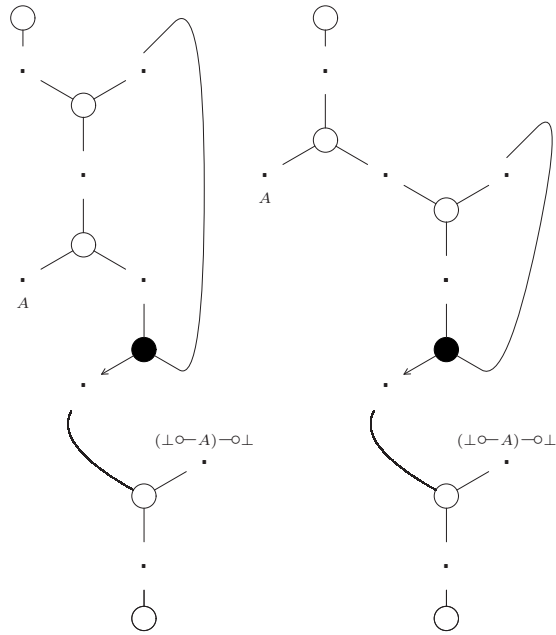


Figure 16: Left identity for par followed by *grishinb*

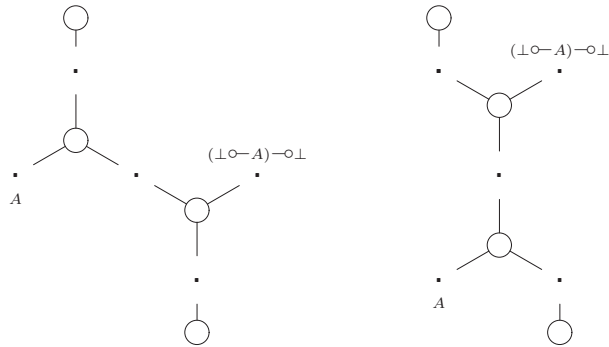


Figure 17: The $0-$ contraction followed by *grishina*

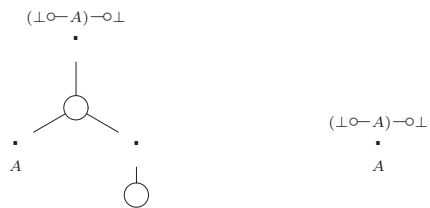
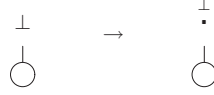


Figure 18: Left identity for par and right identity for tensor

If $d = 1$ then π is one of the axioms. We consider each case separately.

If the conclusion of the sequent is $\perp \vdash \epsilon$ then the corresponding proof structure and abstract proof structure look as shown below.

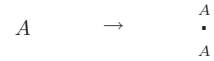


Note how this is a proof net of $\perp \vdash \epsilon$ as required.

Similarly, if the conclusion of the axiom is $\epsilon \vdash \mathbf{1}$ then the corresponding proof structure and abstract proof structure form a proof net of this sequent as shown below.



Finally, if the conclusion of the axiom is $A \vdash A$ for some formula A then we are in the following situation



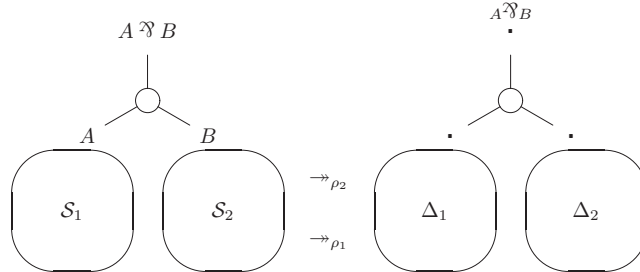
which is a proof net of $A \vdash A$.

If $d > 1$ then we look at the last rule in the proof. Suppose it is a L^{\wp} rule.

$$\frac{\begin{array}{c} \vdots \pi_1 \\ A \vdash \Delta_1 \end{array} \quad \begin{array}{c} \vdots \pi_2 \\ B \vdash \Delta_2 \end{array}}{A \wp B \vdash \Delta_1 \circ \Delta_2} [L^{\wp}]$$

Given that both π_1 and π_2 have a depth smaller than d , we can apply to induction hypothesis to obtain a proof structure S_1 with hypothesis A which reduces to a tensor tree of $A \vdash \Delta_1$ by reduction sequence ρ_1 and a proof structure S_2 with hypothesis B which reduces to a tensor tree of $B \vdash \Delta_2$ by reduction sequence ρ_2 .

We can combine these two proof nets as shown below.

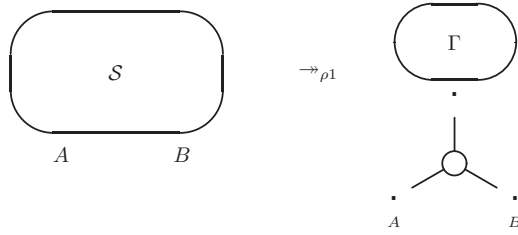


Note that, since ρ_1 and ρ_2 operate on different parts of the resulting abstract proof structure, any interleaving of ρ_1 and ρ_2 will provide a valid reduction sequence ρ producing a proof net of $A \wp B \vdash \Delta_1 \circ \Delta_2$.

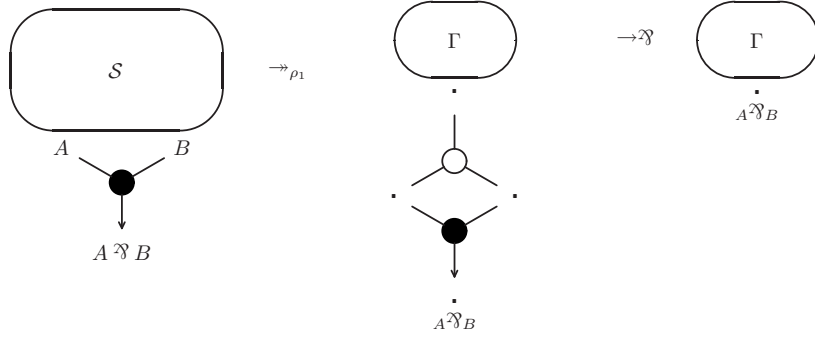
Suppose the last rule is a R^{\wp} rule.

$$\frac{\vdots \pi_1}{\Gamma \vdash A \circ B} [\mathcal{R}^{\wp}]$$

Given that π_1 has a depth of $d - 1$ we can apply the induction hypothesis to give us a proof structure with conclusions A and B which converts to a tensor tree of $\Gamma \vdash A \wp B$ by a conversion sequence ρ_1 .



We can add the par link for $A \wp B$ to the proof structure above after which the reduction sequent ρ_1 produces a redex for the $[\mathcal{R}^{\wp}]$ contraction. We append this contraction at the end of ρ_1 to produce the final contraction sequence ρ , producing a proof net of $\Gamma \vdash A \wp B$ as required.



The other logical rules are similar and easily verified.

\Leftarrow Suppose \mathcal{A} corresponding to \mathcal{S} converts to a tensor tree \mathcal{T} by means of a conversion sequence s . We proceed by induction on the length l of s to construct a sequent proof of \mathcal{T} .

If $l = 0$ then $\mathcal{A} = \mathcal{T}$. We proceed by induction on the number of connectors c in \mathcal{T} .

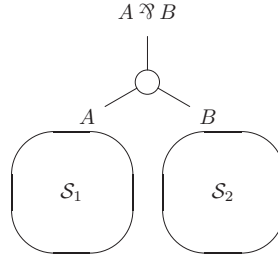
If $c = 0$ then S and T are of the form

$$A \rightarrow \frac{A}{A}$$

which corresponds to the sequent proof

$$\frac{}{A \vdash A} [Ax]$$

If $c > 0$ then by Lemma 8, we know that S has a formula which is the main leaf of its link, call it D . We proceed by case analysis. If D is of the form $A \wp B$, then we are in the following situation.



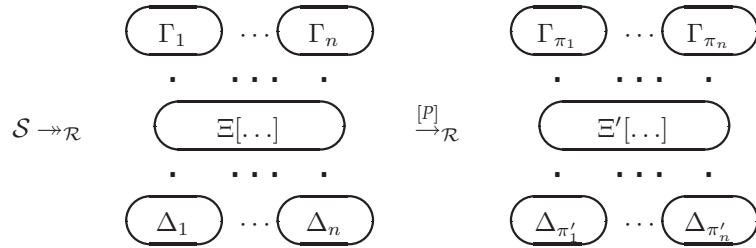
Because the proof structure is a tree, the par link separates the structure into two parts: S_1 with hypothesis A and S_2 with hypothesis B . By induction hypothesis, there are derivations d_1 of $A \vdash \Delta_1$ and d_2 of $B \vdash \Delta_2$, which we can combine as follows.

$$\frac{A \vdash \Delta_1 \quad B \vdash \Delta_2}{A \wp B \vdash \Delta_1 \circ \Delta_2} [L\wp]$$

The other cases are similar.

Suppose now $l > 0$. We look at the last conversion.

Suppose the last conversion is a structural conversion, then we are schematically in the following situation.



Given that we have a structural conversion $[P]$ we know there is at least one structural rule which corresponds to it, where one of the leaves of both Ξ and Ξ' is displayed. In case this leaf is a hypothesis of Ξ (assume it's Γ_i), induction hypothesis gives us a proof d which we can extend as follows.

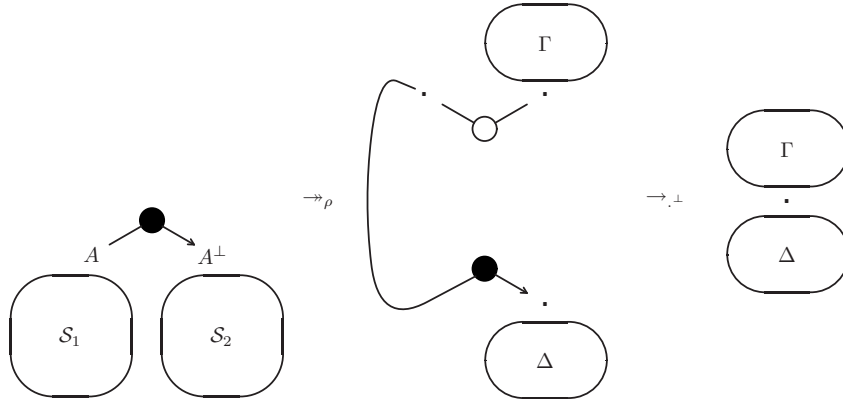
$$\frac{\begin{array}{c} \vdots \\ d \\ \Gamma_i \vdash \Xi \end{array}}{\Gamma_i \vdash \Xi'} [P]$$

In case the leaf is a conclusion Δ_i we operate symmetrically, obtaining.

$$\frac{\begin{array}{c} \vdots \\ d \\ \Xi \vdash \Delta_i \end{array}}{\Xi' \vdash \Delta_i} [P]$$

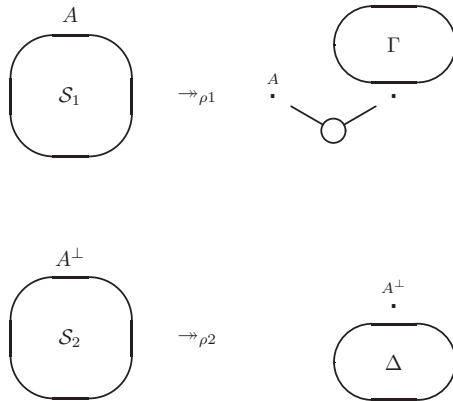
Suppose the last conversion is a contraction. We proceed by case analysis.

$[R.^\perp]$ In case the last conversion is a $^\perp$ contraction we are schematically in the following situation.



Looking backwards from the endsequent, the par link forms a barrier: every structural rewrite has to be performed either fully in Γ — where it will finally end up producing S_1 — or fully in Δ — where it will finally end up producing S_2 . From this perspective, every contraction simply expands a single node and is therefore performed in just one of the two substructures as well. Therefore, we can separate the conversions of ρ into those which are fully in S_1 reducing it to $\Gamma \vdash [A]$ and those which are fully in S_2 reducing it to $A^\perp \vdash \Delta$. We will call these two reduction sequences ρ_1 and ρ_2 respectively.

Removing the par link from the figure above gives us the following two proof structures with their corresponding reduction sequences.

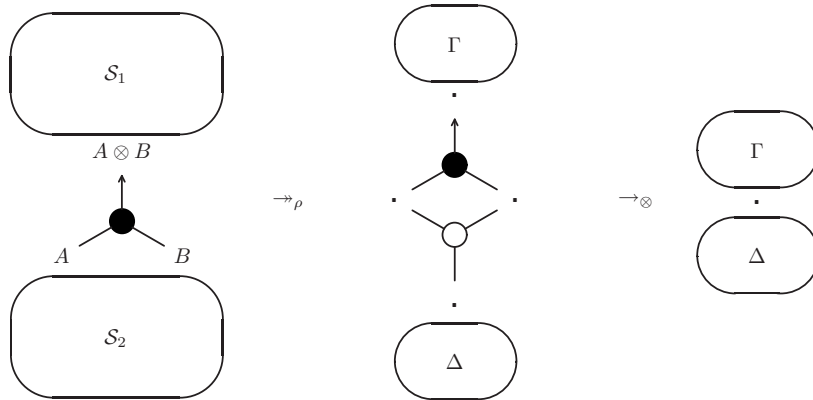


Since the length of $\rho_1 + \rho_2$ is less than the length of ρ — the final contraction being removed — we can apply the induction hypothesis to give us a proof d_1 of $\Gamma \vdash [A]$ and a proof d_2 of $A^\perp \vdash \Delta$. We can combine these two proofs into a proof of $\Gamma \vdash \Delta$ as follows.

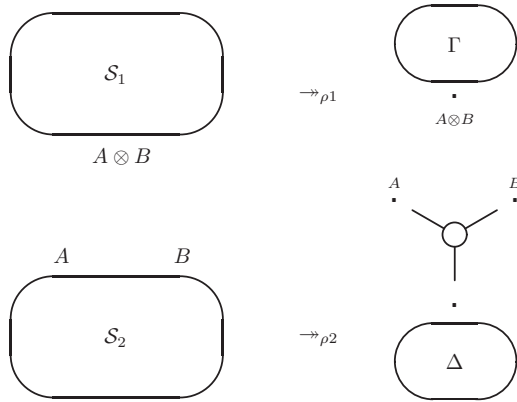
$$\frac{\frac{\vdots d_1}{\Gamma \vdash [A]} \quad \frac{\vdots d_2}{A^\perp \vdash \Delta}}{\Gamma \vdash A^\perp} R.\perp \quad \frac{\Gamma \vdash A^\perp \quad A^\perp \vdash \Delta}{\Gamma \vdash \Delta} Cut$$

$[R^\perp]$ Symmetric.

$[L\otimes]$ In case the last contraction is a \otimes contraction, the proof structure and the conversion sequence for the corresponding abstract proof structure look as shown below.



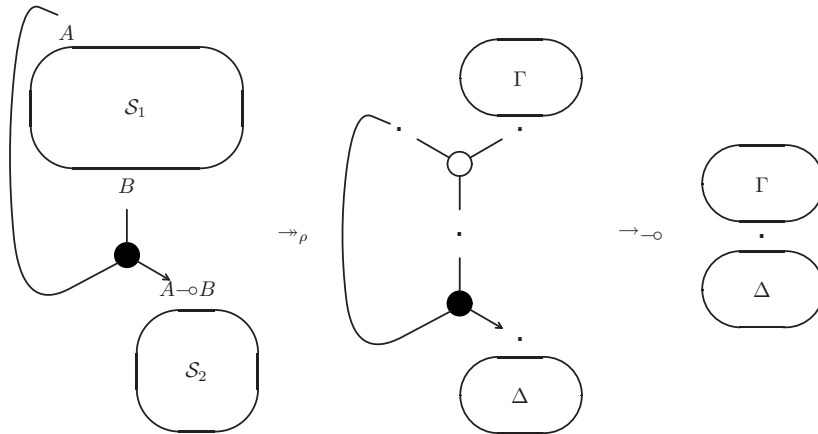
We again eliminate the par link and its contraction and partition the remaining conversions over two disjoint sequences as shown below.



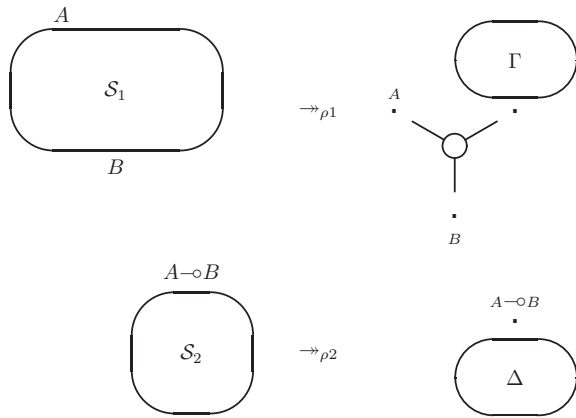
Now the induction hypothesis gives us a derivation d_1 of $\Gamma \vdash A \otimes B$ and a derivation d_2 of $A \circ B \vdash \Delta$.

$$\frac{\frac{\vdots d_1}{\Gamma \vdash A \otimes B} \quad \frac{\frac{\vdots d_2}{A \circ B \vdash \Delta} L \otimes}{A \otimes B \vdash \Delta} Cut}{\Gamma \vdash \Delta}$$

[R- \circ] If the last contraction is a \circ contraction, the proof structure and reduction sequence look as follows.



As before we remove the par link and its contraction and separate the conversion sequences which are in Γ and Δ . The result is shown below.

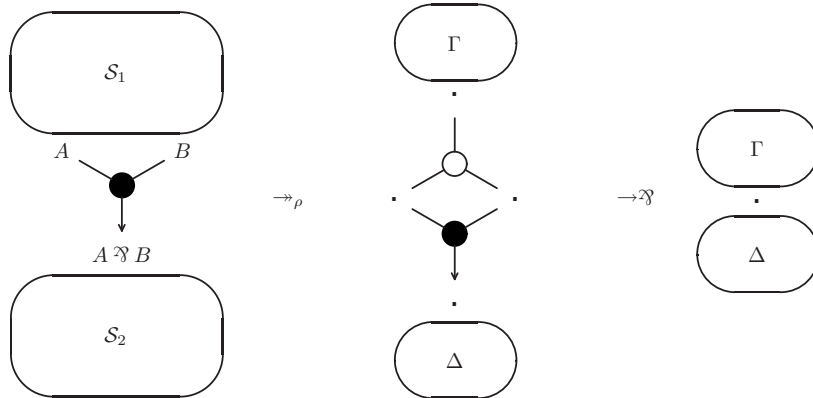


Induction hypothesis now gives us a derivation d_1 from S_1 to $\Gamma \vdash A > B$ and a derivation d_2 from S_2 to $A \multimap B \vdash \Delta$. We can combine these proofs in the following way.

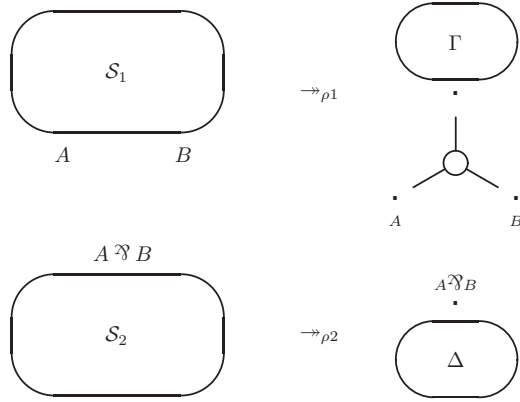
$$\frac{\frac{\frac{\vdots d_1}{\Gamma \vdash A > B}}{\Gamma \vdash A \multimap B} R_{\multimap}}{\Gamma \vdash \Delta} \frac{\frac{\vdots d_2}{A \multimap B \vdash \Delta}}{\Gamma \vdash \Delta} Cut$$

$[R_{\multimap}]$ Symmetric.

$[R_{\wp}]$ If the last contraction is a \wp contraction, the proof structure and reduction sequence look as follows.



Removing the par link and splitting the remaining conversions over the two substructures will give us the situation shown below.



We apply the induction hypothesis to obtain a sequent proof d_1 of $\Gamma \vdash A \circ B$ and a sequent proof d_2 of $A \wp B \vdash \Delta$ and combine these two proofs as follows.

$$\frac{\frac{\frac{\vdots d_1}{\Gamma \vdash A \circ B}}{\Gamma \vdash A \wp B} R\wp}{\Gamma \vdash \Delta} \frac{\frac{\vdots d_2}{A \wp B \vdash \Delta}}{Cut}$$

4 Complexity

In this section I will discuss the computational complexity of the contraction criterion for several different fragments of the proof net calculus.

4.1 Binary Without Structural Conversions

A first case is to decide the contractibility of a proof structure containing only binary links and without any structural conversions. When we look at the redexes of the different contractions, we see that there is no possibility of overlap: a link with three ports cannot be linked at two of its ports by two different links while having each of its nodes be at most once a conclusion and at most once a premiss of its link, as required by our definition of proof nets.

So even a naive contraction strategy which traverses the graph in search of contractible par links and contracts them as soon as they are found then makes another pass until it either fails to contract any par links — in which case the proof structure is not a proof net — or until there are no par links left — in which case we *do* have a proof net. This gives us an $O(n^2)$ algorithm, where n is the number of links in the graph. Without too much effort, we can improve this to $O(p^2)$ where p is the number of par links in the graph.

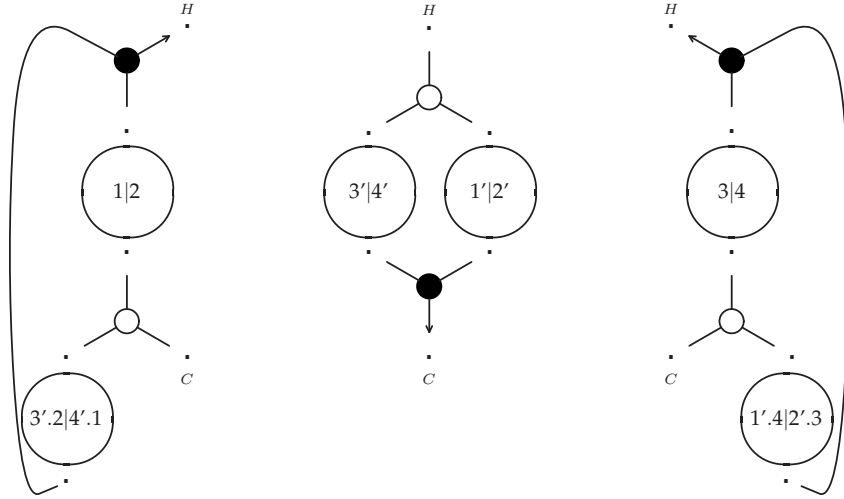


Figure 19: Contractions: binary dual residuated with Grishin

4.2 Binary Grishin

A more complex case uses only the binary connectives but adds the Grishin rules of Figures 13 and 14. Figures 19 and 20 shows the schematic contractions for the binary residuated and dual residuated connectives in this situation, with $A|B$ indicating that either Grishin rule A or Grishin rule B applies, depending on the structure the tensor link finds itself in, and with $A.B$ indicating that we apply Grishin rule A followed by Grishin rule B . We remark that this sequencing operation means moving links first up then towards the C formula in the structure.

A second important point to note is that these operations are *nondeterministic*. For example if both

5 Definability

Defining connectives by using combinations of other connectives. Eg.

$$\begin{aligned} A^\perp &\equiv A \multimap \perp & {}^\perp A &\equiv \perp \multimap A \\ A^1 &\equiv A \multimap \mathbf{1} & {}^1 A &\equiv \mathbf{1} \multimap A \\ \diamond A &\equiv A \otimes \mathbf{1} & \square A &\equiv A \multimap \mathbf{1} \end{aligned}$$

NB these are *behavioral* equivalences, there is no interderivability between any of the proposed formulas in a translation.

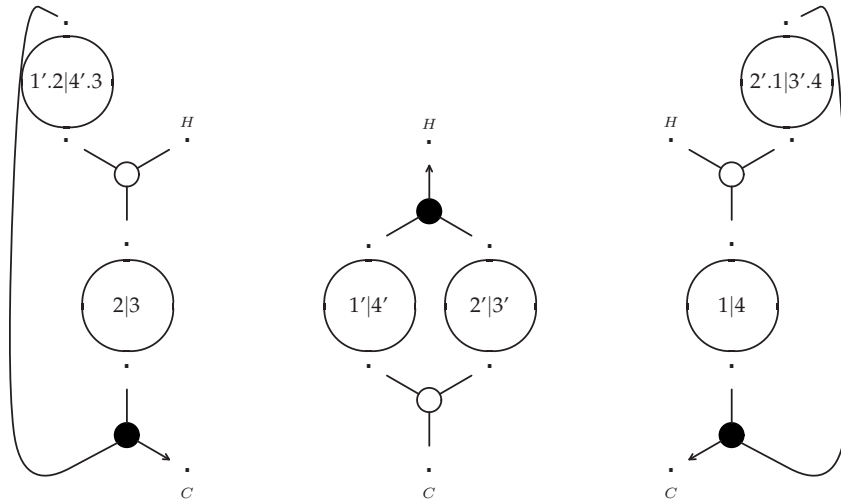


Figure 20: Contractions: binary residuated with Grishin

6 Fragments

TLG (unary/binary residuals)

TLG+Galois (unary/binary residuals+unary Galois)

Lambek Galois (Galois only, implicit associative product)

Bilinear Lambek Calculus (binary res+dual)

Display Logic (all except binary gc/dgc)

Abrusci/Ruet (binary res+dual, negation defined or tensor-only)

7 Conclusions

A Translation Key

References

- Areces, C., Bernardi, R. & Moortgat, M. (2001), Galois connections in categorial type logic, in G.-J. Kruijff, L. Moss & R. T. Oehrle, eds, 'Proceedings of FGMOL 2001', Vol. 53 of *Electronic Notes in Theoretical Computer Science*, Elsevier, pp. 3–20.
- Bernardi, R. & Moortgat, M. (2007), Continuation semantics for symmetric categorial grammar, in Springer, ed., 'Proceedings of WoLLIC 2007', Vol. 4567 of *LNCS*, pp. 53–71.

Here	LL	DL	TLG	BLL
Structural — Nullary				
ϵ		Φ		1
Structural — Unary				
$\langle \cdot \rangle$		\circ	$\langle \cdot \rangle$	
$\lfloor \cdot \rfloor$		\flat	\flat	
$\lceil \cdot \rceil$		\sharp	\sharp	
Structural — Binary				
\circ	,	;	\circ	
$<$		$<$		
$>$		$>$		
\diamond				
\triangleleft				
\triangleright				

Table 6: Translation Key — Structural Connectives

- Girard, J.-Y. (1987), 'Linear logic', *Theoretical Computer Science* **50**, 1–102.
- Goré, R. (1998), 'Substructural logics on display', *Logic Journal of the IGPL* **6**(3), 451–504.
- Lambek, J. (1993), From categorial grammar to bilinear logic, in K. Došen & P. Schröder-Heister, eds, 'Substructural Logics', Oxford University Press, Oxford, pp. 207–237.
- Moortgat, M. (1997), Categorial type logics, in J. van Benthem & A. ter Meulen, eds, 'Handbook of Logic and Language', Elsevier/MIT Press, chapter 2, pp. 93–177.
- Moortgat, M. (2007), Symmetries in natural language syntax and semantics: the Lambek-Grishin calculus, in 'Proceedings of WoLLIC 2007', Vol. 4567 of *LNCS*, Springer, pp. 264–284.
- Moot, R. & Puite, Q. (2002), 'Proof nets for the multimodal Lambek calculus', *Studia Logica* **71**(3), 415–442.

Here	LL	DL	TLG	BLL
Logical — Nullary				
1	1	1		1
\perp	\perp	0		0
Logical — Unary				
\diamond		\diamond	\diamond	
\square		\square	$\square \downarrow$	
1		1	1	l
1		1	1	r
\perp	\perp	0	0	
\perp		0	0	
Logical — Binary				
\otimes	\otimes	\otimes	\bullet	\otimes
\circ	\circ	\rightarrow	\backslash	\backslash
\circ		\leftarrow	$/$	$/$
\otimes	\otimes	\oplus		\oplus
\wp		\vee		
\wp		\wedge		
\leftrightarrow				
\leftrightarrow				
\leftrightarrow				
\leftrightarrow				
\leftrightarrow				
\leftrightarrow				
\leftrightarrow				
\leftrightarrow				

Table 7: Translation Key — Logical Connectives